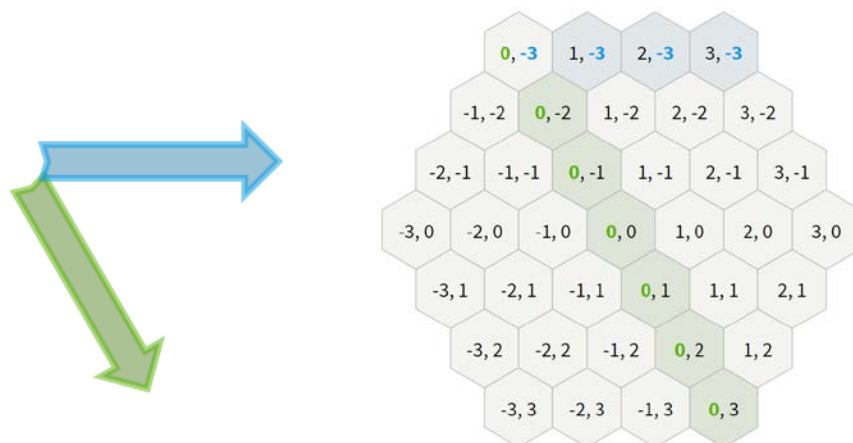


经过讨论，已经基本确定两篇论文的思路和实施方案。

- 1、对于改投 TVCG 的论文，将四边形网格改为六边形，增加移动基站数据，并且将原来的道路模型扩展为基于轨迹的查询模型。同时，在可视化设计上，以矩阵视图体现不同时空上的轨迹可视化效果，地图上以蓝噪声表示数据密度。
- 2、对于 VLDB 论文主要考虑下列问题：移动数据时空粒度不同，如何针对这些数据进行统一查询，在不同粒度的时空 cube 上进行查询，问题和难点，如何解决？

本周已经实现了四边形网格到六边形的改造，六边形的网格划分实现上略难于四边形，但是由于从中心到六个方向的距离相等，因此对道路的覆盖要优于四边形。这里采用了网上一个教程，使用 Axial 坐标。其坐标如下图所示。这里稍微麻烦的是坐标转换，也就是如何将地理空间坐标转换成六边形网格坐标。



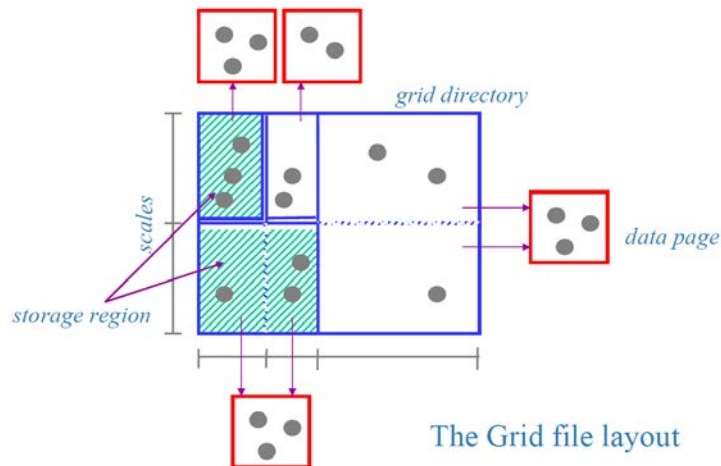
在查询模型上，认真思考了以前算法中存在的主要问题：

- 1、**时间间隔相等。**对于很多移动对象来说，时空轨迹点并非等间隔记录，而且要插值成时间等间隔也较为复杂。能否有方法解决这一问题？
- 2、**查询轨迹时每个网格需要做一次遍历。**查询经过每个网格的轨迹，由于不知道该网格是某一条轨迹的第几个点所在位置，所以需要每个网格进行遍历，查询在指定时间段内的轨迹点。对于繁忙的区域或路段，网格内的轨迹点数量还是较大的，即使按时间排序查询时间仍较长。能否不遍历而读取网格内所有的轨迹点？
- 3、**索引中包含轨迹点的详细信息，导致索引文件较大。**以前为了读取效率直接将轨迹点的详细信息，包括速度、方向、时间、车辆 id 等全部写在索引中。当数据量再增加时，加载索引的时间将非常长，而且索引占据的内存也很大。可不可以只保存最关键的轨迹点信息，如记录 id？
- 4、**索引是否支持分布式计算？**

对于上述问题，我重新设计了索引。本质上说，时空数据查询其实是根据多重属性查询数据。多属性索引应用最广泛的是 GridFile 索引结构。GridFile 将多维空间均匀划分，但是为了保证数据存储的均衡性，将相邻的空间合并存储，同时也避免了不必要的索引空间浪费。但是 GridFile 仍然是对点的索引，并不能很好地应用到轨迹索引上。为了实现轨迹索引，我们希望在尽量小的开销上实现轨迹点的查找。

按照时空特征，将时空划分为均匀大小的三维网格。其中空间按经纬度划分（到地图左上角的距离），时间按自定义的时间最小粒度划分（如小时、天等）。划分后的网格，如果存在轨迹点则用一个 bucket 记录轨迹点索引。Bucket 中每个索引记录包含四项内容：

[key, last, messageId, next]



Last 和 next 分别记录的是当前轨迹点前后时刻的轨迹点索引，即双向指针。key, last 和 next 都是经过 hash 编码后的整数。Hash 函数由空间坐标、移动对象 id 和时间点混合编码，编码后冲突率很低。这样当指定时空范围查询轨迹时，首先找到时空网格，然后读取每个网格内 bucket 记录，再根据双向指针遍历 bucket 记录即可找到所有的轨迹。

总体来说，索引包含两层：时空网格索引（也是 hash 索引）、轨迹双向 hash 索引。由于每个网格可以分别读取，这对于分布式计算是非常方便的，最后只要将这些轨迹索引合并即可。另外，由于索引中只保存 messageId，减少了索引的空间，能够支持大规模的数据查询。

对于这一索引也存在一些问题：

1、数据均衡性

在城市范围内，数据大都集中在繁华地段，数据很难做到均衡。

2、索引空间浪费

由于数据集中在道路上，对于人烟稀少的地方或夜间，数据较少，在实现上可以忽略这些网格内的数据，不产生内存开销。

下周工作

下周主要是将温州出租车数据导入数据库，并且对移动基站数据建立索引进行查询。另外，在建立索引的时候，我做了一些简单的统计，发现网格内的数据非常不均衡。